УДК 004.89

ВИКОРИСТАННЯ ГРУПИ АЛГОРИТМІВ WORD EMBEDDINGS ДЛЯ ПРЕДСТАВЛЕННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ

Туртуріка Віктор

Науковий керівник: канд. ф.-м. наук, доцент Паращук С.Д.

Центральноукраїнський державний педагогічний університет імені Володимира Винниченка, м. Кропивницький, Україна

В статті розглядається застосування сучасної техніки представлення текстової інформації у векторному просторі – word embeddings, для розв'язання задач інтелектуального аналізу тексту. Окрім того, стаття містить детальний опис однієї із реалізацій word embeddings – групи алгоритмів word2vec. До опису роботи word2vec також додається детальний огляд відповідної архітектури штучних нейронних мереж. Також у статті подається порівняння техніки word embeddings із іншими принципами представлення тексту, що застосовуються у галузі інтелектуального аналізу тексту (text mining).

Ключові слова: інтелектуальний аналіз тексту, машинне навчання, нейронні мережі, word embeddings, word2vec, Continuous Bag of Words, Skip-gram model.

The usage of Word embeddings algorithm group for presentation of the text information

V. Turturika

Scientific supervisor: Candidate of Physics and Mathematics Sciences, Docent Parashchuk S.D.

The article deals with the application of modern presentation technique of text information in vector space, that called word embeddings and used for solving problems of text mining. Also, the article contains a detailed description of one of the implementations of word embeddings – word2vec group of algorithms. A description of the work of word2vec also includes a detailed overview of the corresponding architecture of artificial neural networks. Also, the article compares the technology of word embeddings with other principles of presentation of text used in the field of text mining.

Key words: text mining, machine learning, artificial neural networks, word embeddings, word2vec, Continuous Bag of Words, Skip-gram model.

Постановка проблеми. Інтелектуальний аналіз тексту (text mining) – один із напрямів інтелектуального аналізу даних (data mining) та штучного інтелекту, метою якого є отримання інформації з колекції текстових документів

з використанням методів машинного навчання та обробки природної мови. Основними задачами, які вирішує інтелектуальний аналіз тексту є класифікація та кластеризація текстових документів. Класифікація документів поділяється на низку напрямків залежно від розмірності елементів класифікації. Кластеризація документів застосовує засоби машинного навчання без учителя для розподілу документів заданої множини на певну множину класів. Серед інтелектуального аналізу тексту виділяють ряд підзадач, до яких найчастіше відносять такі:

- Інформаційний пошук (information retrieval) – або ідентифікація корпусів тексту (text corpus), що полягає у збиранні та накопичені множини текстових матеріалів із глобальної мережі інтернет, баз даних або інших джерел (наприклад при оцифруванні друкованих носіїв інформації). Метою інформаційного пошуку є підготовка текстової інформації до аналізу.
- Інтелектуальний аналіз тексту частково застосовується у сфері розпізнавання природної мови (natural language processing), а саме для синтаксичного аналізу та розмітки мови (part of speech tagging)
- Розпізнавання іменованих сутностей (named entity recognition) – використовується журналістами або статистиками для автоматизованого визначення іменованих величин, таких як імена відомих людей, найменування фірм та організацій, відомих місць тощо.
- Розпізнавання спеціальних шаблонів – напрям, подібний до named entity recognition, однак замість іменованих сутностей використовуються інші спеціалізовані сутності, такі як номера телефонів, адреси електронної пошти, штрихкоди та інші спеціальні фрази.
- Визначення пов’язаних пар слів (кореферентність – coreference) – ідентифікація іменників та іменникових словосполучень, що позначають один і той самий предмет.
- Емоційний аналіз слів – визначення за поданим текстом його емоційного забарвлення, внутрішнього стану автора, його бачення, думки, експресивності тощо.

Наріжним каменем усіх вищезгаданих задач та підзадач інтелектуального аналізу тексту є представлення власне текстової інформації у пам’яті комп’ютера. Впродовж усього розвитку сфери інтелектуального аналізу тексту представлення тексту зазнавало еволюційних змін. Ці зміни охоплюють період від простого та інтуїтивного унарного представлення (one-hot encoding) до сучасних підходів заснованих на принципах штучних нейронних мереж та машинного навчання із учителем (word embeddings).

Аналіз останніх досліджень та публікацій. Сьогодні сфері інтелектуального аналізу тексту приділяється неабияка увага як серед наукової спільноти, так і серед інженерів програмного забезпечення у передових IT компаніях світу. Досить часто ці спільноти перетинаються і першокласні науковці застосовують на практиці свої знання на посаді Data scientist в тій, чи інший IT компанії. Звичайно, що такий симбіоз науки та бізнесу тільки сприяє загальному розвитку сфери інтелектуального аналізу тексту. Яскравим прикладом такого симбіозу є робота працівника IT-гіганта Google Томаса Міколова *«Efficient Estimation of Word Representations in Vector Space»*[3] в якій було вперше описані техніки *word2vec* із різними варіантами реалізації архітектури мережі (*Continuous Bag of Words* та *Skip-gram model*). Принципи, описані в роботі, сколихнули стрімкий розвиток у сфері інтелектуального аналізу тексту, стимулювали покращення й оптимізацію описаних методів та створення альтернативних груп алгоритмів. Серед таких алгоритмів варто виокремити *GloVe* та *fastText*.

Група алгоритмів *GloVe* або *Global Vectors* є дітищем лабораторії комп’ютерної лінгвістики Стенфордського університету, яка описана у статті *«Global Vectors for Word Representation»*[4]. Ця група алгоритмів використовує сингулярний розклад матриці та принципи машинного навчання із учителем, подібні до принципів *word2vec*. Ще одним значним алгоритмом інтелектуального аналізу тексту для створення векторів word embeddings є результат роботи групи Facebook AI Research Lab (FAIR) – бібліотека *fastText*. Принципи роботи бібліотеки описані у публікації «*Bag of Tricks for Efficient Text*

Classification»[1] одним із співавторів якої є вже вищезгаданий Томас Міколов. На відміну від *word2vec* та *GloVe* алгоритм *fastText* використовує принципи машинного навчання без учителя (unsupervised learning).

Постановка завдання. Метою статті є висвітлення основних принципів роботи векторного представлення (word embeddings) текстової інформації у задачах інтелектуального аналізу тексту, порівняння цього підходу із традиційними методами та аналіз його різних реалізацій.

Традиційні підходи представлення тексту

Одним із найважливіших завдань інтелектуального аналізу тексту є вибір способу представлення тексту в пам'яті комп'ютера. Чим оптимальніше вибране представлення, тим успішнішою та ефективнішою буде робота того, чи іншого алгоритму аналізу тексту та більш репрезентативними будуть результати його роботи.

Мабуть, чи не першою ідеєю, що спадає на думку для представлення тексту на комп'ютері є кодування слів числами за порядком їх розміщення у словнику. Ідея є досить продуктивною у своїй простоті – натуральний ряд є нескінченним і можна занумерувати всі необхідні слова не турбуючись про виникнення колізій. Однак у цієї ідеї є один, досить суттєвий недолік: слова у словнику зазвичай ідуть в алфавітному порядку і при добавленні нового слова потрібно буде перенумерувати більшу частину слів словника. Але найбільшим мінусом цього підходу є повна втрата семантичного значення слова. Наприклад слова «пес», «собака» та «цуценя» суттєво різняться у лексичному сенсі та ще й стоять далеко один від одного у словнику. Однак, у той же час, вони означають самця, самку та дитинча одного і того самого виду тварин тому є близькими у семантичному сенсі. Для того, щоб отримати можливість зберегти семантичну близькість слів було запропоновано використовувати word embeddings, тобто поставити у відповідність кожному слову деякий вектор, що відображає його у певному «просторі сенсів слів». Найпростіше представити слово у вигляді вектора можна наступним чином. Візьмемо простір векторів розмірності, що рівна кількості слів у словнику. Тоді кожне слово буде однозначно

представлене у вигляді вектора, якщо навпроти його позиції в словнику поставити одиницю у відповідній координаті вектора. Таке представлення слів отримало назву One-hot encoding (OHE).

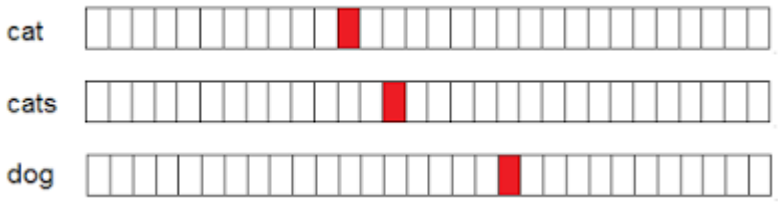

Рис. 1. One-hot encoding

Однак one-hot encoding все ще не може похвалитись представленням семантичного сенсу слова. Проте OHE може бути використане для певного ряду задач інтелектуального аналізу тексту, коли значення одного конкретно взятого слова не таке вже й важливе. Наприклад, візьмемо декілька наборів слів або текстів. Якщо потрібно представити тексти у векторній формі то можна застосувати OHE для кожного окремого слова й отримані вектори додати. Тобто на виході отримаємо підрахунок кількості слів у тексті в одному векторі. Такий підхід називається «мішок слів» (Bag of Words), тому що втрачається вся інформація про взаємозв'язок слів у тексті. Однак використовуючи модель Bag of Words можна вже порівнювати тексти (наприклад розрахувати косинус кута між векторами для двох текстів). Також можна удосконалити модель та представити корпус (набір текстів) у вигляді матриці «слово-документ» (term-document). Таку матрицю часто називають «зворотним індексом» (inverted index) у тому сенсі, що звичайний/прямий індекс виглядає як «документ-слово» і є досить незручним для швидкого пошуку. Матриця «слово-документ» використовується у тематичному аналізі текстів, де її намагаються представити у вигляді добутку матриць «слово-тема» та «тема-документ». В найпростішому випадку із матриці «слово-документ» за допомогою сингулярного розкладу отримують представлення слів через теми і документів через теми [5].

Word embeddings

Вищеописані підходи були (і залишаються) оптимальними для часів (та сфер) в яких кількість текстів є невеликою, а словник обмеженим. Однак із

поширенням всесвітньої мережі інтернет у відкритому доступі з'явилось надзвичайно великі об'єми текстової інформації з якими традиційні підходи втратили свою актуальність. У 2013 році чеським аспірантом Томасом Міколовим був запропонований новий метод для кодування слів у вектори (word embeddings). Цей метод базується на дистрибутивній гіпотезі, запропонованій ще у 1954 році Гаррісом [2]. Згідно цієї гіпотези слова, що зустрічаються в однакових контекстах є семантично близькими. Під контекстом тут розуміється сусідні слова та словосполучення для даного слова. Наприклад у реченні «*Quick brown fox jumps over the lazy dog»* контекстом для слова *«fox»* є словосполучення «*Quick brown*» та *«jumps over»*. Міколов запропонував для кожного слова із словника (причому словник може складатись із значної кількості слів) ставити у відповідність не one-hot encoding вектори (розмірність яких рівна кількості слів у словнику), а певні вектори меншої розмірності (зазвичай від 50 до 1000). Ключовим моментом стало правило: близькі за значенням слова (тобто слова у які зустрічаються у подібних контекстах) мають відповідати подібним векторам (подібність визначається як косинус кута між векторами). Для того, щоб побудувати правило із описаними властивостями був застосований апарат штучних нейронних мереж. У своїй роботі [3] Міколов описав техніку перетворення слів у вектор, яку так і назвав – word2vec. Окрім того Томас запропонував дві архітектури штучних нейронних мереж (*Continuous Bag of Words* та *Skip-gram model*) для своєї техніки.

Таким чином був запропонований новий спосіб представлення текстової інформації для інтелектуального аналізу тексту, для якого великі об'єми вхідних корпусів перетворились із непосильної перешкоди на ефективну перевагу. Адже при більших об'ємах тренувальної множини тільки покращується якість навчання нейронної мережі, що виступає моделлю при такому підході. Окрім того, новий спосіб містив деякі цікаві «побічні ефекти». Оскільки кожному слову відповідає вектор, то для слів стають доступними всі можливі операції над векторами. Класичним (та канонічним) прикладом являється наступний вираз, запропонований самим Міколовим:

[king] – [men] + [woman] ≈ [queen]

Тобто, стає можливим виконувати «додавання та віднімання слів» та отримувати осмисленні результати! Як видно із прикладу, якщо від векторного представлення «короля» відняти «чоловіка» та додати векторне представлення «жінки» то отримаємо вектор, що відповідає слову «королева». Про подібні результати можна навіть і не мріяти, використовуючи традиційні методи представлення тексту.

Continuous Bag of Words та Skip-gram model

Запропонована Міколовим техніка word2vec для створення векторного представлення слів (word embeddings) містила аж два різні варіанти реалізації. Ці варіанти різняться архітектурою штучної нейронної мережі, котра власне і реалізує «функцію перетворення слів у вектори». Обидві нейронні мережі є звичайними мережами прямого поширення із одним вхідним шаром (input layer), одним прихованим шаром (hidden layer) та вихідним шаром (output layer). Мережі різняться представленням вхідного та вихідного векторів, що у свою чергу впливає на навчання та архітектуру.

Принципом роботи мережі Continuous Bag of Words (CBOW) є передбачення слова за його контекстом, а для мережі Skip-gram навпаки – передбачення контексту за вхідним словом.

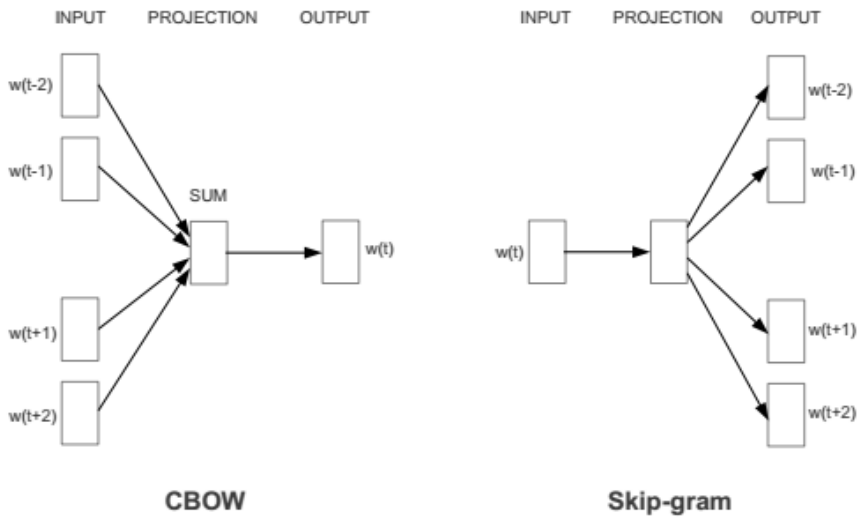

Рис. 2. Архітектури Continuous Bag of Words та Skip-gram model.

Розглянемо кожну архітектуру детальніше.

CBOW – це звичайна «модель мішка слів», процес тренування якої полягає у поданні на вхід контекстів (зазвичай це чотири слова-сусіди: двоє попередніх та двоє наступних) без урахування порядку слідування у тексті. Виходом мережі є найбільш ймовірне слово при даному контексті.

Skip-gram модель, яку ще називають k-skip-n-gram моделю, це послідовності довжини n, в яких елементи знаходяться на відстані не більшій ніж k один від одного. Наприклад маємо речення *«fox jumps over the dog»*. Для нього 1-skip-2-gram будуть виглядати як набори біграм («fox jumps», «jumps over», «over the», «the dog») та наступні послідовності: «fox over», «jumps the», «over dog» – тобто пропускаємо одне (1-skip) слово, а із того, що залишилось справа і зліва від пропущеного створюється біграми (2-gram).

Риторичним залишається питання: для яких же випадків краще використовувати ту, чи іншу архітектуру нейронної мережі? Власне сам Міколов у своїй роботі рекомендує використовувати CBOW для великих корпусів текстів (сто мільйонів, мільярд та навіть більше) оскільки навчання швидше і краще працює із більш частотними словами. Для Skip-gram краще використовувати невеликі корпуси текстів (менше за сто мільйонів слів), оскільки ця модель краще враховує рідкісніші слова та працює повільніше.

Висновки. В ході дослідження було проаналізовано еволюцію методів представлення текстової інформації в галузі інтелектуального аналізу тексту від традиційних статистичних методів (*One-hot encoding, Bag of Word*) до сучасної техніки word embeddings. Детально розглянуто основи техніки представлення тексту word embeddings та описано тонкощі її реалізації. Окрім того, у статті було детально досліджено принципи роботи методу перетворення слів у вектори word2vec та двох його реалізацій CBOW та Skip-gram, що різняться архітектурою нейронних мереж. Виявлено таку суттєву перевагу word embeddings над статистичними методами, як «семантичне додавання/віднімання слів».

Список літератури

1. Bag of Tricks for Efficient Text Classification [Електронний ресурс] / A.Joulin, E. Grave, P. Bojanowski, T. Mikolov – Режим доступу до ресурсу: https://arxiv.org/pdf/1607.01759.pdf.

2. Harris Z. Distributional structure [Електронний ресурс] – Режим доступу до ресурсу: https://link.springer.com/chapter/10.1007/978-94- 017-6059- 1_36

3. Efficient Estimation of Word Representations in Vector Space [Електронний ресурс] / T.Mikolov, K. Chen, G. Corrado, J. Dean – Режим доступу до ресурсу: https://arxiv.org/pdf/1301.3781.pdf.

4. Pennington J. GloVe: Global Vectors for Word Representation [Електронний ресурс] / J. Pennington, R. Socher, C. Manning – Режим доступу до ресурсу: http://www.aclweb.org/anthology/D14-1162.

5. tf–idf [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Tf%E2%80%93idf.